



KỸ THUẬT LẬP TRÌNH

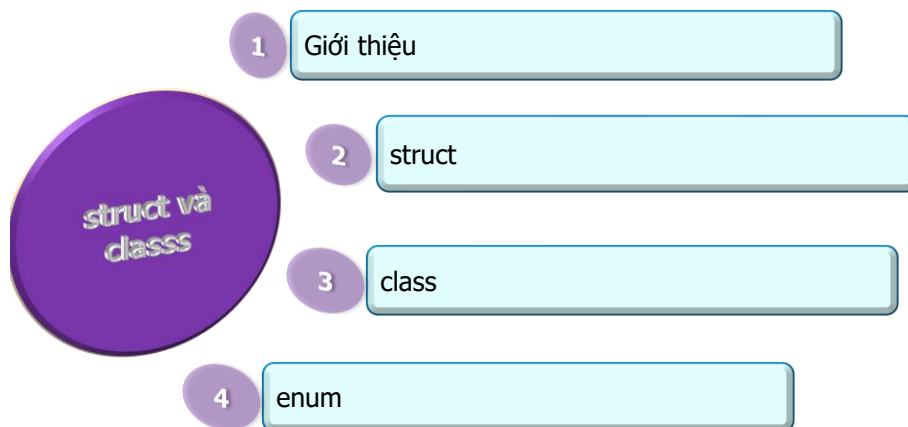
Chương 5: struct và class

Mục tiêu

► Sau khi học xong chương này, người học có thể:

- 1 Hiểu và vận dụng struct, class vào bài toán thực tế
- 2 Phân biệt được các thành phần, phép toán cụ thể với tùy kiểu dữ liệu định nghĩa

Nội dung

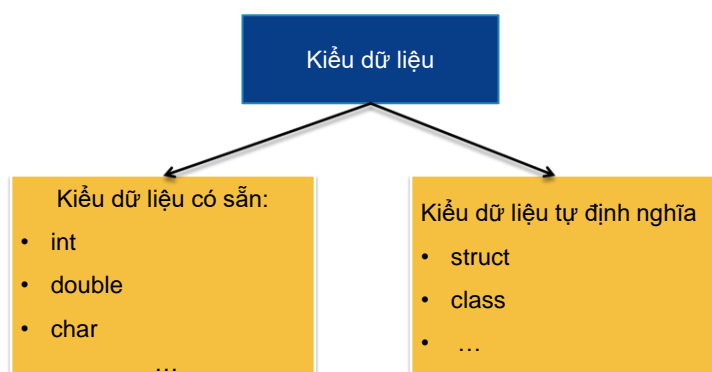


▶ 3

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum



▶ 4

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ Khi nào cần kiểu dữ liệu tự định nghĩa?
- ▶ Kiểu dữ liệu tự định nghĩa giải quyết những vấn đề gì?

▶ 5

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ Định nghĩa struct
- ▶ Truy xuất các thành phần
- ▶ Các phép toán cơ bản
- ▶ Nhập/xuất
- ▶ struct và function
- ▶ struct và array
- ▶ struct trong struct
- ▶ struct và pointer

▶ 6

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► **Định nghĩa struct**

- Struct là kiểu dữ liệu do người lập trình tự định nghĩa. Struct có thể chứa nhiều thành phần dữ liệu có kiểu khác nhau. Mỗi thành phần của struct gọi là thành viên (member) và có tên (memberName).

► Ví dụ:

Struct lưu trữ thông tin của sinh viên.

Các thành viên của struct: họ tên, địa chỉ, tuổi, điểm,

► 7

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► **Cú pháp định nghĩa struct:**

```
struct structName
{
    dataType memberName;
    dataType memberName;
    ...
    dataType memberName;
};
```

Trong đó:

- structName: tên của kiểu dữ liệu được định nghĩa
- dataType: kiểu dữ liệu của từng thành viên của struct
- membername: tên của từng thành viên của struct
- **CHÚ Ý:** kết thúc việc định nghĩa struct phải có dấu ;

► 8

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

- 1. Giới thiệu
- 2. struct

- 3. class
- 4. enum

Ví dụ:

```
struct SinhVien
{
    string hoten;
    string diachi;
    int tuoi;
    double diem;
};
```

► 9

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

- 1. Giới thiệu
- 2. struct

- 3. class
- 4. enum

Khai báo biến kiểu struct:

```
structName variableName;
```

Ví dụ:

```
SinhVien sv;
```

Có thể vừa định nghĩa struct vừa khai báo.

```
struct SinhVien
{
    string hoten;
    string diachi;
    int tuoi;
    double diem;
} sv;
```

► 10

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Truy xuất các thành phần

`variableName.memberName`

Trong đó:

- `variableName`: tên biến có kiểu struct.
- `memberName`: tên thành phần trong struct.

Ví dụ:

```
SinhVien sv1;
sv1.hoten = "Tran Van Minh";
sv1.diachi = "Thanh pho Ho Chi Minh";
sv1.tuoi = 20;
sv1.diem = 7.2;
```

▶ 11

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Các phép toán cơ bản

- **Phép gán**: có thể gán 2 biến struct cho nhau nếu cùng kiểu. Tương ứng với lệnh gán từng thành phần cho nhau.
- **Phép so sánh**: không thể so sánh 2 biến struct với nhau dù cùng kiểu. Chỉ có thể **so sánh từng thành phần**.

▶ 12

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Nhập/xuất

Nhập hoặc xuất từng thành phần của biến kiểu struct.

Ví dụ:

```
SinhVien sv;
cout << "Nhap ho ten sinh vien: ";
getline(cin, sv.hoten);
cout << "Nhap dia chi: ";
getline(cin, sv.diachi);
cout << "Nhap tuoi: ";
cin >> sv.tuoi;
cout << "Nhap diem: ";
cin >> sv.diem;

cout << "Thong tin cua sinh vien la:\nTen: " <<
sv.hoten << "\nDia chi la: " << sv.diachi << "\nTui
la: " << sv.tuoi << "\nDiem la: " << sv.diem << endl;
```

▶ 13

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Bài tập

1. Viết chương trình định nghĩa struct NhanVien gồm các thông tin:

- Mã nhân viên kiểu **string**
- Họ tên nhân viên kiểu **string**
- Năm sinh kiểu **int**
- Chức vụ kiểu **string**
- Bậc lương kiểu **double**

Sau đó khai báo để có thể nhập và xuất thông tin của 1 nhân viên cụ thể.

▶ 14

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

- 1. Giới thiệu
- 2. struct

- 3. class
- 4. enum

Trình tự chương trình đề xuất:

```
//chỉ thị tiền xử lý: #include...

//namespace

//định nghĩa struct NhanVien

//hàm main

{

    //khai báo biến kiểu NhanVien

    //Nhập từng thành phần của biến

    //Xuất thông tin từng thành phần

}
```

► 15

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

- 1. Giới thiệu
- 2. struct

- 3. class
- 4. enum

► struct và function

- Nên định nghĩa các struct trước khi định nghĩa các hàm.
- Struct truyền cho hàm dưới các dạng:
 - Tham trị
 - Tham chiếu
 - Con trỏ
- Hàm có thể trả về kiểu struct.

► 16

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► **Bài tập**

2. Viết chương trình định nghĩa struct `NhanVien` gồm các thông tin:

- Mã nhân viên kiểu `string`
- Họ tên nhân viên kiểu `string`
- Năm sinh kiểu `int`
- Chức vụ kiểu `string`
- Bậc lương kiểu `double`

Sau đó viết hàm nhập và xuất thông tin của 1 nhân viên cụ thể.

► 17

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► **Gợi ý:**

```
void nhap ( NhanVien &nv );
void xuat ( const NhanVien nv );
```

► 18

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Bài tập

3. Viết chương trình định nghĩa `struct PhanSo` gồm các thông tin:

- Tử số kiểu `int`
- Mẫu số kiểu `int`

Sau đó viết các hàm:

- Nhập và xuất thông tin 1 phân số.
- Hàm nhận vào 2 phân số, trả về kết quả cộng của 2 phân số này.
- Hàm main: khai báo phân số `a`, `b`, `c` thuộc kiểu `PhanSo`. Gọi hàm nhập giá trị cho 2 phân số `a` và `b`. Xuất thông tin 2 phân số này. Gọi hàm cộng để trả về kết quả tổng 2 phân số `a` và `b`. Gán kết quả cho phân số `c` và xuất `c`.

► 19

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Bài tập

Gợi ý bài 3:

```

        void nhap (PhanSo &ps);
        void xuat (const PhanSo ps);

PhanSo cong (PhanSo ps1, PhanSo ps2)
{
    PhanSo psKetQua;
    psKetQua.tuso = ...;
    psKetQua.mauso = ...;
    return psKetQua;
};

```

► 20

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Bài tập

4. Phát triển BT3 thành chương trình tính tổng, hiệu, tích, thương của 2 phân số.

▶ 21

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ struct và array

- ▶ struct có thể chứa thành phần là array.
- ▶ Có thể khai báo 1 biến array kiểu struct.

▶ 22

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ struct và array

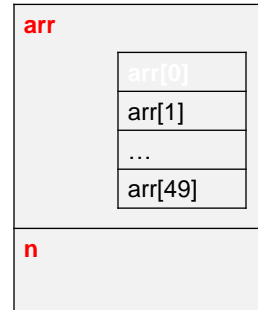
- ▶ struct có thể chứa thành phần là array.

Ví dụ: struct lưu dãy số nguyên

```
const int MAXSIZE = 50;
struct intArr
{
    int arr[MAXSIZE];
    int n;
};
```

Cách truy xuất từng phần tử:

```
intArr a;
a.arr[1] = 3;
a.n = 15;
```



▶ 23

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ struct và array

- ▶ Ví dụ: dựa trên ví dụ struct kiểu dãy số nguyên. Viết các hàm nhập, xuất 1 biến kiểu intArr. Viết hàm trả về tổng các phần tử trong dãy số lưu trữ ở biến kiểu intArr.

▶ 24

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

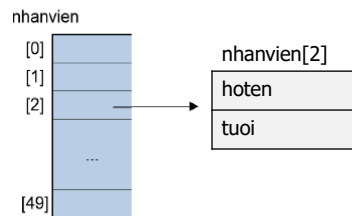
▶ **struct và array**

- ▶ Có thể khai báo 1 biến array kiểu struct.

Ví dụ:

```
struct NhanVien
{
    string hoten;
    int tuoi;
};

NhanVien nhanvien[50];
```



▶ 25

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ **struct và array**

- ▶ Ví dụ: dựa trên ví dụ struct NhanVien. Viết các hàm nhập, xuất thông tin cho 10 nhân viên.

▶ 26

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

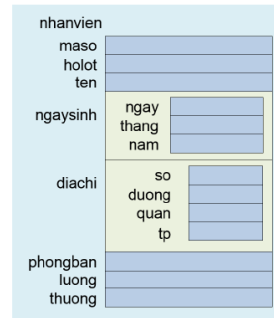
3. class
4. enum

▶ struct trong struct

```
struct NTN
{
    int ngay, thang, nam;
};
```

```
struct DiaChi
{
    string so;
    string duong;
    string quan;
    string tp;
};
```

```
struct NhanVien
{
    int maso;
    string holot;
    string ten;
    NTN ngaysinh;
    DiaChi diachi;
    string phongban;
    double luong;
    double thuong;
};
```



▶ Khai báo: **NhanVien nv;**
 nv.ho = "Tran Van";
 ▶ Truy cập: nv.ten = "Son";
 nv.ngaysinh.ngay = 8;
 nv.ngaysinh.thang = 3;
 nv.ngaysinh.nam = 1980;

▶ 27

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ struct và pointer

- ▶ Có thể khai báo con trỏ kiểu struct. Sau đó cho trỏ đến địa chỉ của 1 biến kiểu struct đã khai báo.
- ▶ Có thể dùng cấp phát động.

▶ Ví dụ: với struct NhanVien đã khai báo

```
NhanVien nv;
```

```
NhanVien *pnv;
```

```
pnv = &nv;
```

Truy xuất thành phần:

```
(*pnv).maso = 5;
```

Hoặc `pnv -> maso = 5;`

▶ 28

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Bài tập struct:

5. Định nghĩa kiểu dữ liệu **SinhVien** gồm các thông tin:

- Họ tên kiểu string;
- Ngày tháng năm sinh kiểu NTN (struct NTN chứa từng thông tin ngày, tháng, năm đều là kiểu số nguyên);
- Điểm toán kiểu số thực.

Viết chương trình:

- Nhập thông tin cho toàn bộ sinh viên của 1 lớp học (tối đa 30 sinh viên). Lưu trữ toàn bộ thông tin 30 sinh viên dưới dạng mảng chứa từng giá trị kiểu SinhVien bằng cấp phát động.
- Xuất lại toàn bộ thông tin của sinh viên nào có điểm toán lớn hơn điểm trung bình của cả lớp.

► 29

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Bài tập struct:

5. Định nghĩa kiểu dữ liệu **Kho** gồm: tên kho, tải trọng.

Tiến hành dùng cấp phát động để tạo ra nơi lưu trữ cho n kho có thông tin kiểu struct Kho ($0 < n < 50$).

Viết hàm:

- + Nhập đầy đủ dữ liệu của n kho này.
- + Xuất lại đầy đủ thông tin n kho.
- + Xuất tên kho nào còn trống. Biết rằng nếu tải trọng > 10 là còn trống.
- + main để kiểm chứng toàn bộ chương trình.

► 30

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ Định nghĩa class
- ▶ Public members
- ▶ Member functions
- ▶ Constructor
- ▶ Header file

▶ 31

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ Định nghĩa class

Tương tự struct nhưng có bao gồm nhiều thành phần trong đó có hàm.

- ▶ Cú pháp khai báo class:

```
class className
{
    memberList;
};
```

Trong memberList:

- Nếu là biến thì khai báo như cú pháp khai báo biến.
- Nếu là hàm thì khai báo như function prototype.
- **LƯU Ý:** với memberlist phải xác định mức độ truy cập.

▶ 32

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Định nghĩa class

- Khi định nghĩa class phải lưu ý đến mức độ truy xuất các member.
- Mức độ truy xuất các member trong class: **private**, **public**, **protected**.
 - **Private**: được liệt kê bên dưới từ **private** và dấu :
Ý nghĩa: các biến liệt kê dạng **private** không thể liệt kê trực tiếp bên ngoài class (muốn truy xuất có thể thông qua member là function).
 - **Public**: được liệt kê bên dưới từ **public** và dấu :
Ý nghĩa: các biến liệt kê dạng **public** có thể được truy xuất trực tiếp từ bên ngoài class.
- Mặc định: **private**

► 33

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Public members

- Với các member là biến đơn thì việc truy xuất tương tự như truy xuất `struct_member`.
- **Member là function**: Khai báo và định nghĩa bên trong class; cũng có thể dùng function prototype để khai báo trong class. Việc thực hiện định nghĩa `function_member` có thể diễn ra bên ngoài class với cú pháp:

```
dataType className::functionName([parameterList])
{
    statements
};
```

- Dấu `::` là toán tử phân giải phạm vi.

► 34

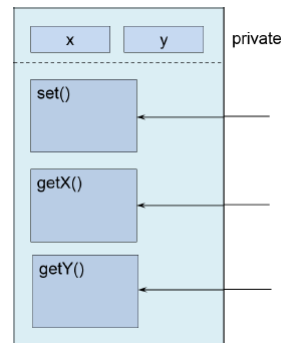
Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Ví dụ: Xây dựng class Point

```
class Point
{
private:
    int x, y;
public:
    int getX();
    int getY();
    void set(int a, int b);
};
```



//getX: trả về giá trị của x

//getY: trả về giá trị của y

//set: gán 2 giá trị số nguyên: a cho x, b cho y.

► 35

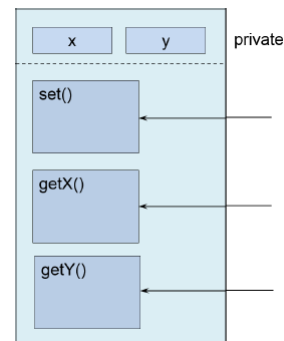
Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Ví dụ: Xây dựng class Point

```
int Point::getX()
{
    return x;
}
int Point::getY()
{
    return y;
}
void Point::set(int a, int b)
{
    x = a;
    y = b;
}
```



► 36

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Ví dụ: Xây dựng class Point bằng cách định nghĩa hàm trực tiếp trong class

```
class Point
{
private:
    int x, y;
public:
    int getX()
    {
        return x;
    }
    int getY()
    {
        return y;
    }
    void set(int a, int b)
    {
        x = a;
        y = b;
    }
};
```

► 37

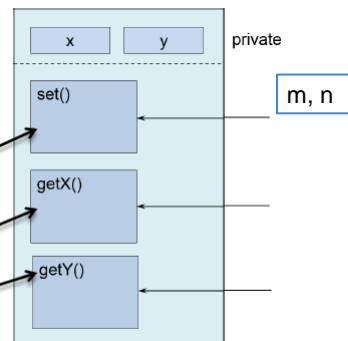
Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► Ví dụ: Xây dựng class Point

```
int main()
{
    Point s;
    int m, n;
    cout << "Nhap hai so nguyen: ";
    cin >> m >> n;
    s.set(m, n);
    cout << s.getX() << endl;
    cout << s.getY() << endl;
}
```



► 38

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

► **Bài tập:**

6. Viết chương trình xây dựng **class PhanSo** gồm:

Private: tử số, mẫu số

Public:

- Hàm trả về giá trị của tử số
- Hàm trả về giá trị của mẫu số
- Hàm thiết lập giá trị cho tử số
- Hàm thiết lập giá trị cho mẫu số
- Hàm tính tích hai phân số.

Sau đó tạo hàm main để kiểm chứng.

► 39

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

```
class phanso
{
private:
    int tuso, mauso;
public:
    int getts();
    int getms();
    void setts(int a);
    void setms(int b);
    phanso tinh(phanso ps1, phanso ps2);
};
```

► 40

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

```
int phanso::getts()
{
    return tuso;
}
```

```
int phanso::getms()
{
    return mauso;
}
```

► 41

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

```
void phanso::setts(int a)
{
    tuso = a;
}
```

```
void phanso::setms(int b)
{
    mauso = b;
}
```

► 42

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

```

phanso phanso::tich(phanso ps1, phanso ps2)
{
    phanso kq;

    kq.setts( ps1.tuso * ps2.tuso );
    kq.setms( ps1.mauso * ps2.mauso );
    return kq;
}

```

► 43

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

```

int main()
{
    phanso ps1, ps2, kq;
    int ts1, ms1, ts2, ms2;

    cout << "Nhap tu so va mau so phan so 1: "; cin >> ts1 >> ms1;
    cout << "Nhap tu so va mau so phan so 1: "; cin >> ts2 >> ms2;

    ps1.setts(ts1);
    ps1.setms(ms1);
    ps2.setts(ts2);
    ps2.setms(ms2);

    kq = kq.tich(ps1, ps2);
    cout << kq.getts() << "/" << kq.getms() << endl;
}

```

► 44

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

Vấn đề tồn tại ???

//Định nghĩa class

```
int main()
{
    phanso ps1, ps2, kq;
    cout << ps1.getts() << endl;
}
```

//Sẽ xuất giá trị rác

- Vậy ta cần khởi tạo giá trị cho các biến private trong class.
- class hỗ trợ **khởi** tạo giá trị tự động cho các biến private, gọi là **constructor**

► 45

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- **Constructor:** là hàm tự động thực hiện khi 1 đối tượng thuộc class được tạo.
- Constructor dùng để khởi tạo các giá trị cho biến private trong class.
- **Constructor:**
 - Trùng tên với class và không có kiểu.
 - Mỗi class có 1 constructor mặc định: constructor không có tham số.
 - Mỗi class nên có ít nhất 1 constructor có tham số (đều trùng tên với class nhưng số tham số khác nhau).

► 46

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

- 1. Giới thiệu
- 2. struct

- 3. class
- 4. enum

► Constructor:

Ví dụ:

```
class phanso
{
private:
    int tuso, mauso;
public:
    phanso() { tuso = 0; mauso = 0; };
    phanso (int a, int b);
    ~phanso(); //huy constructor
    int getts();
    int getms();

    void setts(int a);
    void setms(int b);
    phanso tinh(phanso ps1, phanso ps2);
};
```

► 47

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

- 1. Giới thiệu
- 2. struct

- 3. class
- 4. enum

► Constructor:

Ví dụ:

```
phanso::phanso(int a, int b)
{
    setts(a);
    setms(b);
}
```

//khi đó trong hàm main:

```
phanso ps1, ps2, kq;
cout << ps1.getts() << endl;
```

//kết quả sẽ là 0

► 48

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Header file

- Tạo file kiểu header file (.h) ở folder header file của project.
- Định nghĩa class ở header file.
- Ở file .cpp sử dụng class thêm chỉ thị:

```
#include "tên_file_header.h"
```

▶ 49

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Bài tập

7. Định nghĩa class PhanSo bao gồm các biến private là tuso, mauso; các hàm public là cong, tru, nhan, chia, ucln, rutgonps.

Sau đó viết chương trình kiểm chứng.

▶ 50

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ Định nghĩa enum
- ▶ Các phép toán
- ▶ Nhập/xuất
- ▶ Mảng 2 chiều và kiểu liệt kê

▶ 51

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ Định nghĩa enum

Là kiểu dữ liệu do người dùng định nghĩa đơn giản nhất, gọi là kiểu liệt kê, dùng để lưu trữ một tập giá trị hằng được đặt tên giúp chương trình dễ đọc và tránh lỗi.

Cú pháp định nghĩa enum:

```
enum typeName {value1, value2,...};
```

Trong đó:

- typeName: tên kiểu dữ liệu
- value1, value2, ... tên của các giá trị, đặt giữa { và }
- Giá trị mặc định theo thứ tự value là 0, 1, ...

▶ 52

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Ví dụ:

```
enum Thu {hai, ba, tu, nam, sau, bay, chunhat};

Thu ngay;

ngay = tu;

cout << ngay << endl;

//kết quả: 2
```

▶ 53

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

▶ Các phép toán

- ▶ Phép gán: gán giữa các biến cùng kiểu enum với nhau, gán giá trị cho biến kiểu enum.
- ▶ Chuyển đổi kiểu dữ liệu khi xuất giá trị lưu trữ: câu lệnh `static_cast<DataType>(variableName)`

▶ 54

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ **Các phép toán**
- ▶ Không thể gán giá trị kiểu int cho biến enum.
- ▶ Chỉ có thể so sánh giữa biến kiểu enum với giá trị.

▶ 55

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ **Các phép toán**
- ▶ **Không sử dụng phép toán số học**

```
DenGiaoThong x, y;
```

```
x = XANH;  
y = x + 1; //error  
x++;      //error
```

- ▶ **Muốn tăng hay thay đổi phải chuyển đổi kiểu dữ liệu**

```
x = static_cast<DenGiaoThong>(x + 1);
```

▶ 56

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

1. Giới thiệu
2. struct

3. class
4. enum

- ▶ **Nhập/xuất:** gián tiếp thông qua biến kiểu enum đã khai báo và cú pháp lựa chọn trường hợp để gán dữ liệu.
- ▶ **Mảng 2 chiều và kiểu liệt kê:** tận dụng chỉ số mảng là chỉ số giá trị kiểu liệt kê do enum trả về.

▶ 57

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018

Q & A

▶ 58

Kỹ thuật lập trình - ĐH Mở TpHCM 12/23/2018